

## 저연산 적응형 확장 블록 탐색 움직임 추정 기법

최수우<sup>1</sup>, 윤종호<sup>1</sup>, 조태경<sup>2</sup>, 최명렬<sup>1\*</sup>

<sup>1</sup>한양대학교 전자전기제어계측공학부, <sup>2</sup>상명대학교 정보통신공학과

# Low Computational Adaptive Expanded Block Search Motion Estimation Method

Su-Woo Choi<sup>1</sup>, Jong-Ho Yun<sup>1</sup>, Tae-Kyung Cho<sup>2</sup> and Myung-Ryul Choi<sup>1\*</sup>

<sup>1</sup>Dept of EECI, Hanyang University

<sup>2</sup>Dept of IEC, Sangmyung University

**요약** 본 논문에서는 영상 내의 움직임은 시간 공간적으로 높은 상관관계를 가지는 특성을 고려하여 움직임 추정을 실행하는 블록을 참조 프레임과의 상관관계에 따라 평균 움직임 블록(AMB)과 부분 움직임 블록(LMB)으로 구분한다. 수정된 고속 탐색 패턴을 통해 적은 연산량을 가지며 블록의 구분, 하위 블록 사용 및 정합도 계산 시 인접 블록을 포함하는 블록 확장(BE)으로 움직임 추정의 정확도를 향상시키는 방법을 제안한다. 실험 결과를 통해서 제안된 알고리즘이 평균적으로 전역 탐색의 7.5%의 연산량으로 전역 탐색 보다 0.5dB, 다이아몬드 탐색 에서 보다 1.7dB의 화질의 향상을 가져오는 것을 확인 하였다. 제안한 알고리즘은 영상 압축이나 프레임률 변환(FRC)에 적용 가능하다.

**Abstract** In this paper, Low Computational Adaptive Expanded Block Search Motion Estimation Method is proposed. Proposed method classifies ME blocks as Average Motion Block(AMB) and Local Motion Block(LMB) according to correlation of reference frame. It could reduce the computational complexity with performing Modified Fast Search(MFS). And accuracy of MV is also increased by 4 sub-blocks on LMB and Block Expansion(BE). The experimental results show that the proposed method has better performance that increased 1.8dB than Diamond Search and 0.6dB than Full Search with 7.5 % computation of Full Search. The proposed method could be applied to video compression and Frame Rate Conversion(FRC).

**Key Words** : ME, Image Processing

### 1. 서론

일반적으로 비디오 영상은 연속되는 프레임 사이에 시간적으로 많은 중복성을 포함하고 있다. H.264, MPEG-4 등의 비디오 압축 표준에서는 시간적 중복성의 제거를 위해 블록 정합(Block Matching) 기반의 움직임 추정 기법(Motion Estimation Algorithm)을 사용한다. 블록 정합은 현재 프레임을 동일한 블록들로 나누고, 오차가 최소인 블록을 참조 프레임의 탐색 영역에서 찾는 꺾법이다. 정합 블록과 움직임 추정을 수행하는 현재 블록의 거리를 움직임 벡터 (Motion Vector)라 한다.

전역 탐색(Full search)은 일반적인 블록 정합 기법이

다. 탐색 영역의 모든 위치를 탐색하여 정확한 움직임 벡터를 찾을 수 있으나 연산량이 많다는 단점이 있다[1]. 전역 탐색의 단점을 극복하기 위하여 연산량을 낮추는 고속 탐색 기법들이 제안 되었다. 3단계 탐색 알고리즘 (Three Step Search : TSS), 새로운 3단계 탐색 알고리즘 (New Three Step Search : NTSS), 4단계 탐색 알고리즘 (Four Step Search : FSS), 다이아몬드 탐색 알고리즘 (Diamond Search : DS) 등이 있다[2-6]. 기존의 고속 탐색 기법은 탐색 영역의 특정 위치만을 비교하므로 추정된 움직임 벡터의 정확도가 낮아질 수 있다. 이에 반해 H.264에서와 같은 여러 형태의 움직임 추정 블록의 사용은 움직임 추정의 정확도를 향상시킨다. 이는 더욱 신뢰

\*교신저자 : 최명렬(hanyangrfd@yahoo.co.kr)

접수일 10년 01월 08일

수정일 (1차 10년 02월 05일, 2차 10년 03월 25일)

게재확정일 10년 04월 09일

성 있는 움직임 벡터를 추정하고 H.264의 압축률 증대의 큰 기여를 한 부분이나 연산량이 이전의 압축표준보다 크게 증가하게 된 이유이기도 하다. 이러한 연산량의 증가는 실시간 처리에 있어서는 많은 부담을 가지는 부분이다. 본 논문에서는 영상 내의 움직임은 시-공간적으로 높은 상관관계를 가지는 특성을 고려하여 움직임 추정을 실행하는 블록을 조건에 따라 2가지로 구분하고 수정된 고속 탐색 패턴을 통해 적은 연산량을 가지며 블록의 구분, 하위 블록 사용 및 인접 블록의 활용으로 움직임 추정의 정확도를 향상시키는 방법을 제안한다.

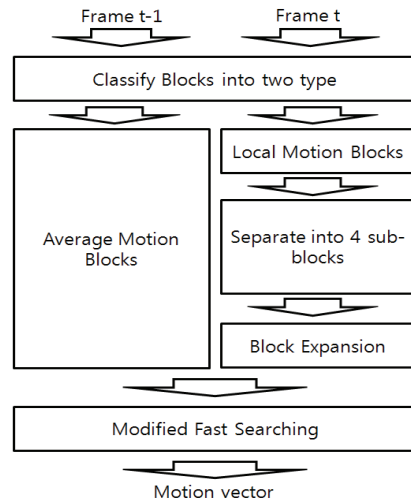
본 논문의 구성은 다음과 같다. 2장에서는 제안한 알고리즘에 대하여 설명하였으며, 3장에서는 기존의 방법들과 성능을 비교하여 제안한 알고리즘의 성능을 검증하였다. 마지막으로 4장에서는 본 논문의 결론을 도출하였다.

## 2. 저연산 적응형 확장 블록 탐색 움직임 추정 기법

본 논문에서는 정확한 움직임 추정을 수행하고 연산량을 줄일 수 있는 움직임 추정 방법을 제안하였다. 영상 내의 움직임은 시간 공간적으로 높은 상관관계를 가지는 특성을 이용하여 움직임 추정을 실행하는 블록을 조건에 따라 2가지로 구분, 프레임 내에서 비슷한 움직임을 가지는 인접한 블록들은 묶여 동시에 탐색하여 연산시간을 줄이고 블록 내에 2가지 이상 서로 다른 움직임을 가지는 블록은 세부 블록으로 나누어 전체 움직임 추정에서 움직임 벡터의 정확도를 높이는 것을 중심으로 접근하였다.

영상내의 객체의 형태는 움직임 추정을 수행하는 단위 블록의 경계와 일치하는 경우가 적으며 움직임 추정 블록 안에 서로 다른 방향을 가지는 2개 이상의 움직임을 갖게 되는 경우에는 기존의 일반적인 알고리즘의 적용 시 움직임 벡터의 정확도를 떨어뜨리는 경우가 발생하는 문제점이 있다. 이런 문제점의 개선을 위해 프레임에서 블록으로 나눌 때 각 블록들은 2가지로 구분하는데 각 명칭을 Average Motion Block(AMB), Local Motion Block(LMB)로 정한다. 연산량의 감소를 위하여 기존 DS 패턴을 수정하여 적용하였으며 LMB에서는 정확한 움직임 추정을 수행하기 위해 4개의 하위 블록을 새로 구성하고 4개의 움직임 벡터의 추정을 수행한다. 제안한 알고리즘은 가변크기 블록 움직임 추정을 지원하는 H.264와 같은 동영상 압축 표준이나 MC-FRC (Motion Compensated Frame Rate Conversion)와 같은 움직임 프레임 올 변환 시 프레임 보간에 사용되어 질 수 있다.

그림 1은 제안 알고리즘의 수행 흐름도를 나타낸다. 제안 알고리즘의 수행은 영상에서의 인접한 두 프레임, 'Frame t-1, Frame t', 입력으로 받게 되고 움직임 추정을 하기에 앞서 'Frame t'는 N x N 크기의 겹치지 않은 블록들로 나누어지며 각 블록의 정합을 위한 참조 블록은 'Frame t-1'에 위치하게 된다. 'Frame t'의 나누어진 블록들은 초기 블록 정합 값에 따라 구분되며 이때 초기 블록 정합의 위치는 'Frame t-1'에서 이미 계산된 움직임 벡터들의 평균값을 사용한다. Average Motion Block(AMB)로 구분된 블록은 Modified Fast Search (MFS)를 수행하게 되고 블록의 움직임 벡터를 결정하게 된다. Local Motion Block(LMB)로 구분된 블록은 다시 하위 4개의 N/2 x N/2 크기의 블록으로 나뉘어져 각각의 블록에 Block Expansion(EB)을 수행하여 움직임 벡터를 결정하게 된다.



[그림 1] 제안된 알고리즘의 수행 흐름도

### 2.1 블록 정합도 계산 방법

블록 정합 시 정합의 정도를 평가하는 방법으로는 SAD(Sum of Absolute Difference), MAE(Mean Absolute Error), MSE(Mean Square Error)등이 일반적으로 사용된다. 각 수식의 계산값이 작을수록 정합도가 크게 되며 본 논문에서는 곱셈과 나눗셈 연산을 하지 않는 SAD를 연산량의 장점을 이유로 사용하였으며 그 식은 다음과 같다.

$$SAD(i, j) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (1)$$

### 2.2 움직임 추정 블록의 구분

움직임 추정 블록의 구분에 있어서 이전 프레임인 ‘Frame  $t-1$ ’에서 추정된 각 블록들의 움직임 벡터의 평균을 구하여 사용하는데 이때 평균 움직임을 Average Motion Vector( $\overline{AMV}$ )라 정의하고 다음의 식을 사용한다.

$$\overline{AMV} = \frac{1}{K} \sum_{i=0}^{K-1} MV(frame_{t-1}(i)) \quad (2)$$

위 식에서 K는 ‘Frame  $t-1$ ’에서 구해진 총 움직임 벡터의 수이고  $\overline{MV}$ 는 움직임 벡터를 말하며  $i$ 는 인덱스를 의미한다.

움직임 추정 블록의 위치를 중심으로  $\overline{AMV}$  위치의 참조 블록과의 블록 정합도를 계산하여 계산된 값으로 블록을 구분하며 SAD값을 사용한다.

AMB, LMB의 구분은  $\overline{AMV}$  위치의 SAD 값이 임계값(Th) 보다 작으면 AMB로 크거나 같으면 LMB로 구분하며 LMB는 움직임 벡터의 정확도의 향상을 위하여 4개의 하위 블록으로 나누게 되며 임계값(Th)은 다음의 식으로 결정된다.

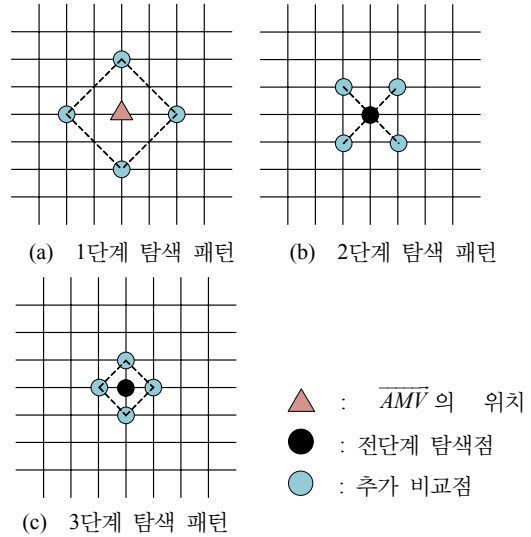
$$Th = \frac{N^2 \cdot I_{max}}{50} \quad (3)$$

여기서 N은 블록의 크기를 말하며  $I_{max}$ 는 한 픽셀이 가지는 최대값을 말한다.

### 2.3 수정된 고속 탐색(Modified Fast Search)

AMB는  $\overline{AMV}$  위치의 SAD값이 적은 부분으로  $\overline{AMV}$ 의 근처에서 최소 정합 오차 블록을 찾는 확률이 높다. 이에 착안하여 MFS는 중심부분에서 작은 움직임에 높은 성능을 보이는 다이아몬드 패턴을 채용하였으며 탐색영역의 외각 탐색 시 움직임 추정의 정확도를 향상시키기 위하여 크로스 패턴을 2단계로 적용 하였다. MFS에 사용되는 패턴은 그림 2과 같은 3가지 패턴을 사용한다. 그림 2의 (a)는 최초 탐색으로  $\overline{AMV}$ 의 위치를 기준으로 변위 2인 상하 좌우의 4개 비교점에서 SAD값을 비교하게 된다. 그림 2의 (b)는 2단계 탐색 패턴으로 전 단계 탐색 점을 기준으로 (-1, 1), (-1, -1), (1, -1), (1, 1) 위치의 4개의 비교점을 추가 SAD값을 비교한다. 그림 2의 (c)는 마지막 3단계로 전 단계 탐색점을 기준으로 변위가

1인 상하 좌우의 4개의 추가점을 비교한 후 최소 SAD값을 가지는 위치를 움직임 벡터로 결정하게 된다.

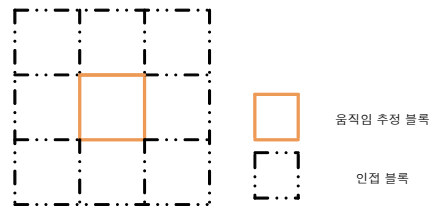


[그림 2] MFS 탐색 패턴

LMB는  $\overline{AMV}$ 와는 상관성이 없는 움직임을 가지는 블록으로 탐색의 시작이  $\overline{AMV}$  위치가 아니라 탐색영역의 중심점에서부터 시작된다.

### 2.4 블록 확장 (Block Expansion)

LMB에서 수행 되는 BE는 움직임을 추정하는 블록에 인접한 블록들이 영상에서 공간적으로 상관성을 가지고 있음에 착안하여 블록 정합 오차 계산 시 인접한 블록으로 확장하여 계산한다. 그림 3과 같은 8개의 인접한 LMB의 활용으로 다양한 형태의 블록형태의 확장을 통해 추정되는 움직임 벡터의 신뢰도를 향상시킬 수 있다[7].



[그림 3] 사용 되는 후보 인접 LMB의 범위

인접한 8개의 LMB중 블록 정합에 사용되는 블록은 움직임 추정 블록과의 상관도에 따라 결정되며 상관도의 판단은 다음과 같은 식이 사용된다.

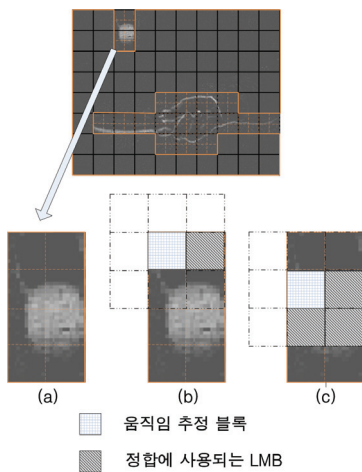
$$\left| SAD_{CB}(\overline{AMV}) - SAD_{UNB}(\overline{AMV}) \right| \leq V_{th} \quad (4)$$

$$V_{th} = \frac{SAD_{CB}(\overline{AMV})}{100} \quad (5)$$

여기서  $SAD_{CB}$ 는 움직임 추정 블록의 SAD 값을  $SAD_{UNB}$ 는 인접 LMB의 SAD값을 말한다.  $V_{th}$ 는 임계값으로  $\overline{AMV}$  위치에서의 현재 블록과 인접 LMB블록의 차의 절대값이 임계값( $V_{th}$ )보다 작거나 같은 인접 LMB를 움직임 추정 시 블록 정합에 사용한다. 그림 4는 SAD값 계산을 같이 하게 되는 확장 블록의 예시를 나타낸다. 그림 4의 (a)는 LMB 부분의 확대 그림을 나타내고 (b)와 (c)는 각 LMB의 움직임 추정 시 정합을 같이 하게 되는 확장 블록을 나타낸다. LMB의 움직임 벡터의 탐색은 그림 4의 (b), (c)와 같이 여러 형태의 블록의 조합으로 이루어지게 된다.

### 3. 시뮬레이션 내용 및 결과

본 논문에서 제안한 알고리즘에 대한 성능 평가는 연속된 영상에 대하여 전역 탐색(Full Search : FS) 알고리즘, 다이아몬드 탐색(Diamond Search : DS)알고리즘, 마지막으로 본 논문에서 제안한 방법을 적용하여 얻은 결과를 비교하여 검증하였다.



[그림 4] 블록 확장 예시

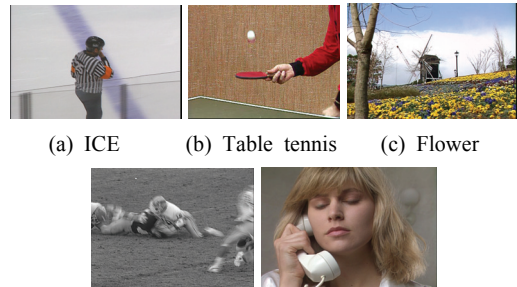
움직임 추정의 정확도를 평가하기 위해서 추정된 움직임 벡터로 보간한 프레임과 원본 프레임의 PSNR (Peak

Signal to Noise Ratio)을 비교하였고 한 블록에서 탐색한 비교점의 평균을 통하여 연산량을 비교함으로써 알고리즘을 평가하였다. FS 및 DS 및 제안한 알고리즘의 MFB의 크기는 16x16, 탐색영역의 크기는 수직 수평으로 -16 ~ +16으로 실험 하였다.

제안한 알고리즘의 성능 평가를 위하여 실험한 이미지는 Ice, Table Tennis, Flower, Football, Susie의 6개의 영상으로 그림 5과 같다. 각각의 영상에 대해 연속되는 25~30프레임을 사용하였다. 컬러(RGB 포맷) 영상은 휘도(Y) 값을 추출하여 휘도(Y) 채널에서 움직임 추정을 하였으며 사용한 변환 식은 다음과 같다.

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (6)$$

그림 5의 (a)는 Ice 영상으로 카메라가 심판을 따라 오른쪽으로 움직이고 있으며 상대적으로 심판을 제외한 부분은 모두 왼쪽의 같은 방향으로 이동하는 영상이다. (b)는 Table Tennis 영상으로 초기 라켓과 손목 부위의 작은 움직임과 공의 빠른 움직임이 함께 하는 영상이다. (c)는 Flower 영상으로 카메라가 일정속도로 오른쪽으로 이동하여 전체적으로 왼쪽으로 움직임을 가지고 있으며 풍차의 날개 부분이 회전하는 영상이다. (d)는 Football 영상으로 카메라는 움직임이 거의 없고 전체적인 움직임이 빠르고 복잡한 영상이다. (e)는 눈 깜빡임과 빠른 고개 젓힘으로 얼굴 부분이 회전하는 움직임을 가지는 영상이다.



(a) ICE (b) Table tennis (c) Flower  
(d) Football (e) Susie

[그림 5] 실험 영상

#### 3.1 PSNR(Peak Signal to Noise Ratio) 비교

PSNR은 서로 다른 두 영상에 대한 차이를 수치나 단위, 정량적으로 판단하기 위한 값으로 값이 클수록 두 영상의 차이가 적음을 나타내며 다음의 식으로 값을 구한다.

$$PSNR = 10 \log_{10} \left( \frac{255 * 255}{MSE} \right) \quad (7)$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (|I(i, j) - K(i, j)|^2) \quad (8)$$

여기서 m, n은 이미지의 가로, 세로의 크기를 i, j는 픽셀의 위치를 나타내는 인덱스를 나타내며 I, K는 비교하는 두 영상의 픽셀값을 의미한다.

[표 1] PSNR의 비교 테이블

이미지	전역탐색	다이아몬드 탐색	제안기법
Ice	28.9	27.5	29.8
Tennis	28.8	26.9	29.2
Flower	22.8	21.5	23.1
Football	34.8	33.7	35.1
Susie	29.4	28.8	30.2
<b>Average</b>	<b>28.9</b>	<b>27.7</b>	<b>29.4</b>

표 1을 보면 제안한 알고리즘에서의 LMB의 블록의 세분화에 의해 전 영상에서 평균적으로 전역 탐색 보다 0.5(dB), 다이아몬드 탐색 에서 보다 1.7(dB)의 화질의 향상을 가져오는 것을 확인할 수 있다.

### 3.2 연산량의 비교

서로 다른 사이즈인 이미지들의 연산량을 비교하기 위해 하나의 블록에서 움직임 벡터를 계산하기 까지 비교하게 되는 평균 탐색점의 수를 확인 하였으며 자세한 내용은 표 2와 같다. 전역 탐색은 탐색 영역내의 모든 비교점들을 탐색 하므로 수직 수평으로 -16~ +16크기의 탐색 영역에서는 모두 일정하게 1089개의 탐색점에서 SAD의 계산이 필요하다. 다이아몬드 탐색은 평균적으로 전역 탐색의 2.5% 정도의 탐색으로 연산량을 줄였으나 움직임 벡터의 정확도가 많이 떨어지는 결과를 확인 할 수 있었다. 제안한 알고리즘은 연산량은 전역 탐색의 7.5%정도로 전역 탐색과 비슷하거나 약간의 화질향상을 나타내는 것을 확인 할 수 있다. 그러나 Flower 및 Football 영상과 같이 부분적인 움직임이 많은 영상, 즉 LMB이 많을 때의 연산량의 증가 부분의 개선이 차후 필요함을 알 수 있다.

[표 2] 연산량 비교 테이블

이미지	전역탐색	다이아몬드 탐색	제안기법
Ice	1089	33.4	66.1
Tennis	1089	14.3	28.2
Flower	1089	20.9	101.2
Football	1089	25.6	182.8
Susie	1089	37.9	69.3
<b>Average</b>	<b>1089</b>	<b>27.7</b>	<b>82.72</b>

## 4. 결론

프레임 보간을 위한 움직임 추정을 y행함에 있어 가장 중요한 관점은 정확한 움직임 벡터를 찾는 것이다. 또한 많은 연산량이 필요한 과정이므로 실시간 처리를 위하여 적은 연산량으로 움직임 추정이 수행되어야 한다. 본 논문에서 제안한 알고리즘은 수정된 고속 탐색 패턴을 통해 적은 연산량을 가지며 블록을 구분, 하위 블록 사용 및 인접 블록의 활용으로 움직임 추정의 정확도를 향상시킨다. 실험 결과를 통하여 제안된 알고리즘이 기존의 전역 탐색 움직임 추정과 다이아몬드 탐색보다 높은 PSNR을 나타내며, 상대적으로 적은 연산량으로 움직임 추정을 수행함을 보였다.

향후 연구 방향으로 LMB이 많을 때의 연산량의 증가 부분의 개선과 인접 블록들간에 상관도의 판단의 정확도를 높일 수 있는 방안에 대한 추가적인 연구가 필요하다. 또한 움직임 추정으로 통해 구한 움직임 벡터를 이용하여 프레임 보간 시 효과적으로 적용하는 방안에 대한 연구를 진행 할 계획이다.

## 참고문헌

- [1] F. Dufaus, et al., "Motion estimation techniques for digital TV: A review and a new contribution," proceedings of the IEEE, Vol. 83, No. 6, pp. 858-876, 1995.
- [2] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans. Communications, Vol. 38, No. 7, pp. 950-953, 1990.
- [3] R. Li, et al., "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technology, Vol. 4, No. 4, pp. 438-442, 1994.
- [4] S. Zhu, et al., "A new diamond search algorithm for



fast block matching motion estimation," IEEE Trans. Image Processing. Vol. 9, No. 2, pp. 287-290, Feb. 2000.

- [5] L. M. Po, et al., "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technology, Vol. 6, No. 3, pp. 419-423, Aug. 1996.
- [6] C. H. Cheung, et al., "A novel cross-diamond search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technology, Vol. 12, No. 12, pp. 1168-1177, Dec. 2002.
- [7] 최명렬, 광통일 외 3명, "프레임 보간을 위한 프레임 차이 기반의 적응형 확장 블록 움직임 추정", 한국통신학회 논문지, 제33권 제8호, pp. 598-604, 2008. 8.

**최 수 우(Su-Woo Choi) [준회원]**



- 2008년 2월 : 한양대학교 전자컴퓨터공학부 (공학사)
- 2008년 3월 ~ 현재 : 한양대학교 전자전기제어계측공학과 석사과정

<관심분야>  
이미지 프로세싱, SoC/ASIC 설계

**윤 종 호(Jong-Ho Yun) [준회원]**



- 2001년 2월 : 한양대학교 전자컴퓨터공학부
- 2003년 8월 : 한양대학교 일반대학원 전자전기제어계측공학과 졸업 (공학석사)
- 2003년 9월 ~ 현재 : 한양대학교 일반대학원 전자통신전파공학과 박사과정

<관심분야>  
SoC/ASIC, 3D, 영상처리

**조 태 경(Tae-Kyung Cho) [종신회원]**



- 1984년 2월 : 한양대학교 전자통신공학과
- 1986년 2월 : 한양대학교 대학교 전자통신공학과 석사
- 2001년 8월 : 한양대학교 대학원 전자통신공학과 박사
- 2003년 9월 ~ 현재 : 상명대학교 정보통신공학과 교수

<관심분야>  
유 · 무선통신프로토콜, 정보보호, 이터닝

**최 명 렬(Myung-Ryul Choi) [정회원]**



- 1983년 2월 : 한양대학교 전자공학과
- 1985년 2월 : 미시간주립대학교 컴퓨터 공학 졸업 (공학석사)
- 1991년 2월 : 미시간주립대학교 컴퓨터 공학 졸업 (공학박사)
- 1992년 9월 ~ 현재 : 한양대학교 컴퓨터공학과 교수

<관심분야>  
SoC/ASIC 설계, 3D, FPD 컨트롤러, RFID응용